CLAIMS

What is claimed is:

1. A computer system for simulating the operation of an embedded system using a software application residing in the memory of a computer, comprising:

a design application configured to form a hardware specification of an embedded system, the design application including:

a design language having at least one graphical symbol adapted to form a finite state machine (FSM) representation of electronic hardware, each graphical symbol of the design language having a graphical portion and a user-definable textual portion defining the behavior of the graphical symbol,

a library including at least one instruction set accurate simulator for simulating the behavior of a target processor core having at least one memory read pin, at least one write memory pin, and at least one interrupt pin,

a graphical user interface adapted to permit a user to select an instruction set accurate simulator of a target processor core and to form a FSM representation of a hardware component in the design language, and

a configuration interface for coupling memory read/write signals and interrupt signals of the instruction set accurate simulator to corresponding signals of a the hardware component;

a test bench builder for generating a graphical representation of at least one interactive test bench and for selecting signals or variables to be coupled to a graphical representation of a user interface for each interactive test bench;

a software debugger including a debugging interface for associating a breakpoint of execution a graphical symbol of the finite state machine representation of the hardware component;

a graphical database and a behavioral database coupling data associated with the hardware description to a code generator;

a compiler for compiling the output of the code generator into an object file of the hardware description;

a discrete event simulation engine configured to execute the object file of the hardware description;

a first API interface adapted to couple the discrete event simulation engine to each of the interactive test benches;

at least one API interface adapted to couple the discrete event simulation engine to the instruction set accurate simulator of the hardware description; and

at least one API interface adapted to couple the instruction set accurate simulator to a software debugger, the software debugger configured to load at least one binary program code compiled for the target processor.

2. The computer system of claim 1, wherein the debugging interface is configured to permit the at least one compiled binary program code to be stepped to the break points of execution.

3. The computer system of claim 1, wherein the design language is an adaptation of the specification and description language (SDL).

4. The computer system of claim 1, wherein the computer system has a simulation speed selected so that the software application executes at a rate greater than one million instructions per second.

5. The computer system of claim 1, wherein the computer system serves as a hardware emulator and has a simulation speed sufficient to boot an embedded system operating system is less than one minute.

6. The computer system of claim 1, wherein the computer system is coupled to a server of a computer network.

7. The computer system of claim 6, wherein the computer network is selected from the group consisting of: a wide area network, a local area network, an Intranet, an extranet, or a computer network coupled to the Internet.

8. The computer system of claim 6, wherein a client computer may be coupled to the computer system by a network connection.

9. The computer system of claim 6, further comprising: a design repository coupled to the server adapted to store a design specification for each of a plurality of embedded systems, each design specification including sufficient information to generate the hardware specification for the embedded system.

10. The computer system of claim 9, further comprising a matching engine coupled to the design repository for associating metadata with each design specification and matching vendors to the stored design specifications based on the metadata.

11. The computer system of claim 9, wherein each design specification includes at least one design specification source file configured to permit the hardware specification to be viewed in the design language and at least one executable file to run a

simulation of the embedded system that includes at least one of the interactive test benches.

12. The computer system of claim 9, wherein each design specification includes at least one dynamic link library (DLL) that may be loaded by a loader executable to permit the design specification to be viewed in the design language and to run a simulation of the embedded system that includes at least one of the interactive test benches.

13. The computer system of claim 9, further comprising: a design manager configured to control access to the design specifications stored in the design repository.

14. The computer system of claim 6, wherein each design specification includes a design specification source file to describe each hardware element system, a file for each instruction set simulator of the embedded system, and a dynamic link library to execute the design specification.

15. The computer system of claim 9, wherein the server is coupled to the client computer via the Internet.

16. The computer system of claim 15, further comprising: an on-line bulletin board coupled to the server for a user to post a design specification.

17. The computer system of claim 16, wherein the bulletin board is a bidding board.

18. The computer system of claim 9, further comprising: content and a content viewer configured to interactively explain to a user at least one attribute of a selected embedded system.

19. The computer system of claim 1, further comprising: a walkthrough application including content and a content viewer configured to interactively explain to a user at least one attribute of an executable file of the hardware description of a selected embedded system.

20. The computer system of claims 18 or 19, wherein the walkthrough application is configured to allow the user to control the operation of a simulation of the embedded system through the content viewer.

21. The computer system of claim 20, wherein the embedded system includes a processor core and at least one reference hardware peripheral for evaluating the processor core.

22. A computer-implemented method for assisting a user to evaluate at least one component of an embedded system, the method comprising:

forming a virtual embedded system including an instruction set accurate simulator of a target processor core coupling read, write, and interrupt signals with a finite state machine (FSM) simulation of at least one hardware element;

coupling a virtual test bench emulating a human/machine interface to the virtual embedded system, the virtual test bench having a graphical user interface adapted to interact with the virtual embedded system;

receiving from the user a binary program code of a software application compiled for the target processor core; and

responsive to a request from a user, loading the software application and running a simulation of the embedded system.

23. The method of claim 22, further comprising:

creating content for explaining the operation of the virtual embedded system; and

responsive to a user request, providing the content as an interactive demonstration.

24. The method of claim 22, wherein the virtual embedded system resides on a server and the method further comprises:

forming a data connection between the server and a client computer;

receiving command and data user inputs from a web browser residing on the client computer; and

providing graphical display data of the simulation running on the server to the web browser residing on the client computer.

25. The method of claim 24, wherein the server is a network server coupled to the client computer via the Internet and the method further comprises:

monitoring the user's interaction with the virtual embedded system; and

recording data associated with an interaction of the user with the virtual embedded system.

26. The method of claim 24, further comprising:

forming a plurality of virtual embedded systems in a library;

receiving a request from a user for a selected virtual embedded system; and

recording data associated with usage of the selected virtual embedded system.

27. The method of claim 26, further comprising:

requesting the user to submit registration data.

28. The method of claim 25, further comprising:

providing each user with a graphical user interface to modify the virtual embedded system;

monitoring modifications made by a plurality of users to the virtual embedded system; and

determining a statistically popular modifications to the virtual embedded system.

29. The method of claim 24, wherein the software debugger is adapted to load binary program code compiled for a target processor core and the method further comprises:

responsive to a user request, loading onto the virtual embedded system a binary executable program file of a software application compiled for the processor core; and

displaying a simulation of the virtual embedded system executing the software application.

30. The method of claim 22, further comprising:

providing the user a graphical interface having a design language including a plurality of graphical symbols adapted to form a finite state representation of a user-defined hardware element that may be coupled to the virtual embedded system.

31. The method of claim 30, wherein the graphical symbols have a graphical portion and a textual portion with at least one of the graphical symbols including a textual portion adapted for the user to describe a behavior of the symbol.

32. The method of claim 31, wherein a user may input a text portion including at least one command written in the ANSI C/C++ language.

33. The method of claim 31, wherein the design language is an adaptation of a specification and description language (SDL).

34. The method of claim 22, wherein a plurality of virtual embedded systems are stored in a database, the method further comprising:

responsive to a user request, selecting one of the virtual embedded systems for evaluation.

35. The method of claim 22 further comprising:

forming content for the embedded system, the content including a at least one graphical control interface configured to interact with the simulation of the embedded system:

responsive to a user request displaying content associated with the embedded system; and

responsive to a user input to the graphical control interface, displaying an interaction with the simulation.

36. The method of claim 35, wherein the content is a tutorial of the embedded system.

37. In a computer system having a graphical interface and a design language for forming a finite state machine (FSM) representation of a hardware partition of an embedded system, a method of designing an embedded system, the method comprising:

forming a library of processors including an instruction set accurate simulator for each of the processor cores in the library;

responsive to a first sequence of user commands, selecting at least one of the processor cores from the library as a target processor;

responsive to a second sequence of user commands, forming a virtual embedded system including an instruction set accurate simulator of a target processor core coupling read, write, and interrupt signals with a finite state machine (FSM) simulation of at least one hardware element;

responsive to a request from the user, loading an executable binary file of a software application compiled for the target processor;

executing a simulation of the virtual embedded system running the software application;

responsive to a user request, displaying a graphical representation of the execution of the software on the virtual embedded system that includes a software debugger interface to debug the loaded software and a virtual test-bench having a graphical user interface adapted to interact with the simulation.

38. The method of claim 37, wherein the design language includes a plurality of graphical symbols with each graphical symbol having a graphical semantic portion and a textual semantic portion.

39. The method of claim 38, wherein the design language includes:

a start object defining a starting point of the finite state machine at an initialization time, the start object having an output connector activated when the start object is initialized;

a task object including a field for inputting computer code in the C language for defining a behavior of the task object and a connector port for coupling the task object to other objects;

a state object for representing a state of the finite state machine;

a decision object having an evaluation field for directing a flow of execution based on a result of an expression in the decision field;

a signal-out object for sending a communication signal;

a signal-in object for receiving a communication signal;

a connector object for connecting control flow;

a symbol object having at least one user-definable pin connector and containing a block or process object;

a block object for describing the behavior of one or more processes; and

a process object for representing a finite state machine process.

40. The method of claim 37, further comprising:

storing the virtual embedded system as a design in a design repository coupled to a server; and

providing access privileges to the design to a selected individual or group.

41. The method of claim 40, further comprising:

responsive to a user command, providing access privileges to the design to a group of vendors.

42. The method of claim 41, wherein the design is accessible from an on-line bidding board.

43. The method of claim 41, wherein access privilege to the design are provided to a group of vendors selected by the user.

44. The method of claim 40, wherein the design is accessible by an embedded system design team.

45. The method of claim 40, further comprising a design manager application, the method further comprising:

permitting one or more members of a design team to select edit privileges of at least one version of the design.

46. The method of claim 45, further comprising:

permitting one or more members of the design team to load and execute a binary program exeucutable of a software application compiled for the target processor core onto the design.

47. The method of claim 45, further comprising:

providing version control and regulating editing access to maintain a consistent version of the design.

48. The method of claim 37, further comprising:

storing a plurality of virtual embedded systems in a library; and

responsive to a user request, providing the user a copy of one of the virtual embedded systems stored in the library.

49. The method of claim 38, further comprising the graphical user interface is configured to permit a a user to associate a breakpoint of execution with a graphical symbol, the method further comprising:

responsive to a user input, associating a breakpoint of execution with a graphical symbol;

receiving a request to debug software; and

stopping the simulation responsive to a command flow of the FSM representation of the hardware element reaching the graphical symbol of the breakpoint of execution.

50. The method of claim 49, further comprising:

responsive to a user request, single-stepping the simulation to sequential breakpoints of execution in the command flow of the FSM representation of hardware elements.

51. The method of claim 49, further comprising:

responsive to a user request, single-stepping the simulation by a pre-selected number of time units.

52. A computer implemented method of embedded system design, the method comprising:

selecting an instruction set accurate simulator of a target processor core;

generating a virtual hardware component that is a finite state representation of at least one hardware component;

linking read, write, and interrupt signals of the instruction set accurate simulator of the target processor core with corresponding signals of the at least one virtual hardware component to form a virtual embedded system;

coupling a virtual test bench to at least one signal or variable of the virtual embedded system to simulate a human/machine interface; and

coupling a software debugger to the virtual embedded system that is configured to load and run on the virtual embedded system at least one binary program executable of a software application compiled for the target processor core.

53. The method of claim 52, further comprising:

selecting the target processor core from a library having a plurality of instruction set accurate simulators for a plurality of processor cores.

54. The method of claim 52, further comprising:

selecting the virtual hardware component from a library of virtual hardware components.

55. The method of claim 54, further comprising:

modifying the virtual hardware component.

56. The method of claim 52, further comprising:

loading benchmark software in an evaluation phase of an embedded system project and running a simulation of the virtual embedded system executing the benchmark software.

57. The method of claim 52, further comprising:

loading binary program executables of development software compiled for the target processor core in a development phase of an embedded systems project and running a simulation of the virtual embedded system executing the development software.

58. The method of claim 57, further comprising:

debugging the development software using a software debugger.

59. The method of claim 52, further comprising:

storing the virtual embedded system as a design having at least one executable file in a design repository.

60. The method of claim 59, wherein the design is stored on a server accessible to a user-group.

61. The method of claim 60, wherein the user-group is a geographically distributed embedded system project team.

62. The method of claim 59, further comprising:

providing a version of the design to a vendor offering a good or service related to the virtual embedded system.

63. The method of claim 62, wherein the design is stored on a server and a vendor is provided access to the design via a network connection.

64. A method of designing an embedded system, the method comprising:

defining a system architecture of the embedded system;

designing a virtual prototype of the embedded system having an instruction set accurate simulator of a target processor core coupling read, write, and interrupt signals with a finite state machine (FSM) representation of at least one hardware element;

coupling the virtual prototype to a software debugger having a debugging interface and a virtual test bench having a graphical representation of a human/machine interface for interacting with the embedded system;

developing at least one software application for the processor core;

loading compiled binary program code of the at least one software application for execution on the virtual prototype; and

initiating a simulation of the virtual prototype executing the at least one software application.

65. The method of claim 64, further comprising:

developing a hardware implementation using the virtual prototype as a functional specification describing a hardware partition.

66. The method of claim 65, further comprising:

evaluating the operation of the embedded system executing the at least one software application; and

responsive to a result of the evaluation, modifying a hardware or software component of the embedded system.

67. The method of claim 64, wherein the step of defining the system architecture comprises:

selecting at least one embedded system component for evaluation;

forming a virtual evaluation platform including the selected embedded system component;

loading a benchmark software application for execution on the virtual evaluation platoform; and

evaluating performance of the virtual evaluation platform executing the benchmark software application.

68. A method of providing information to potential suppliers for procuring a good or service associated with an embedded system, the method comprising:

defining a system architecture of the embedded system;

designing a virtual prototype of the embedded system having an instruction set accurate simulator of a target processor core and a virtual hardware element that is a finite state machine (FSM) representation of a hardware element configured to couple memory read/write requests and interrupt signals with the instruction set accurate simulator of the target processor core;

coupling the virtual prototype to a virtual test bench having a graphical representation of a human/machine interface for interacting with the embedded system;

publishing the virtual prototype as a functional specification from which a vendor may initiate a simulation of the operation of the embedded system.

69. The method of claim 68, wherein the virtual prototype is stored on a computer readable medium and publishing the virtual prototype comprises sending the computer readable medium to a vendor.

70. The method of claim 68, further comprising a network server, wherein the virtual prototype is published by posting the virtual prototype as a design hosted on a database coupled to the network server.

71. The method of claim 70, wherein the virtual prototype is published to a bulletin board of the database.

72. The method of claim 71, wherein the bulletin board is a bidding board.

73. The method of claim 70, wherein the virtual prototype is published to a database having a matching engine.

74. A computer-implemented method for a vendor to acquire information for the procurement of a good or service related to an embedded system project, the method comprising:

accessing a database of virtual prototypes of embedded systems, each of the virtual prototypes having a processor simulator, a finite state machine representation of hardware peripherals, and a virtual test bench emulating a human/machine interface for interacting with a simulation of the operation of the virtual prototype;

instantiating an instance of one of the virtual prototypes; and

evaluating the virtual prototype.

75. The method of claim 74, further comprising:

submitting a quote for a good or service related to the embedded system simulated by the virtual prototype.

76. The method of claim 74, wherein the virtual prototype is configured to show a parts list.

77. The method of claim 74, wherein the virtual prototype is configured to show a component net list.

78. The method of claim 74, wherein the database of virtual prototypes is hosted on a network server accessible by a client computer.

79. A computer-implemented method for evaluating at least one component of an embedded system, the method comprising:

forming a virtual embedded system including an instruction set accurate simulator of a target processor core coupling read, write, and interrupt signals with a finite state machine (FSM) representation of a hardware element;

coupling a virtual test bench emulating a human/machine interface to the virtual embedded system, the virtual test bench having a graphical user interface adapted to interact with the virtual embedded system;

hosting the virtual embedded system on a network server accessible from a client computer via the Internet;

uploading from the client computer a binary executable program file of a software application compiled for the target processor core; and

responsive to a request from a user, running a simulation of the operation of the virtual embedded system executing the uploaded software application.

80. The method of claim 79, further comprising:

forming content for explaining the operation of the embedded system; and

responsive to a request from a user, sending the content to the client computer.

81. The method of claim 79, further comprising:

responsive to a request from the client computer, forming a persitent data connection between the network server and a client computer;

receiving command and data user inputs from a web browser residing on the client computer; and

providing graphical display data of the simulation running on the server to the web browser residing on the client computer.

82. The method of claim 79, further comprising:

monitoring the user's interaction with the virtual embedded system; and

recording as marketing data at least one interaction of the user with the virtual embedded system.

83. The method of claim 79, further comprising:

forming a plurality of virtual embedded systems in a library;

receiving a request from a user for a selected virtual embedded system; and

recording data associated with usage of the selected virtual embedded system.

84. The method of claim 83, further comprising:

requesting the user to submit registration data and associating the registration data with the marketing data.

85. The method of claim 84, further comprising:

providing the user a graphical interface having a design language including a plurality of graphical symbols adapted to form a finite state representation of a user-defined hardware element that may be coupled to the virtual embedded system;

monitoring modifications made by a plurality of users to the virtual embedded system; and

determining a popular modification to the virtual embedded system.

86. A computer program product for simulating an embedded system, the computer program product comprising:

a computer readable medium:

a processor simulator module stored on the medium having an instruction set accurate simulator for at least one processor core;

a design application module stored on the medium for designing a virtual prototype of the embedded system having an instruction set accurate simulator of at least one processor core coupling read, write, and interrupt signals with a finite state machine (FSM) representation of at least one hardware element;

a simulation engine module for executing a simulation of the virtual prototype;

a software debugger module stored on the medium for a user to load compiled program code for execution on the virtual prototype; and

a test bench module stored on the medium for forming a virtual test bench having a graphical representation of a human/machine interface for interacting with an embedded system prototype.

87. A computer program product for providing information on the operation of an embedded system, comprising:

a computer readable medium;

a hardware description of the virtual embedded system stored on the medium, the hardware description including an instruction set accurate simulator of a processor core coupling read, write, and interrupt signals with a finite state machine (FSM) represention of at least one hardware element;

a simulation engine module stored on the computer readable medium for forming a simulation of the embedded system running a software application on the hardware description; and

a test bench module stored on the medium for forming a virtual test bench emulating a human/machine interface to the simulation of the embedded system, the virtual test bench having a graphical user interface adapted to interact with the simulation of the embedded system.

88. A computer program product for evaluating an embedded system, comprising:

a computer readable medium;

at least one executable file of a hardware description of a virtual embedded system stored on the medium, the virtual embedded system including an instruction set accurate simulator of a processor core coupling read, write, and interrupt signals with a finite state machine (FSM) simulation of at least one hardware element;

a simulation engine module for running the at least one executable file as a simulation of the embedded system; and

a test bench module stored on the medium configured to form a virtual test bench emulating a human/machine interface to the virtual embedded system, the virtual test bench having a graphical user interface adapted to interact with the virtual embedded system